MCS/CS 401, Spring 2020
March 11-13 material

March 15, 2020

- activity-selection problem (page 415)
- dynamic programming approach (page 416)
- how to simplify the dynamic programming algorithm by always picking the activity finishing first (pages 417-418, including Theorem 16.1 and its proof)
- the resulting greedy algorithm (subsection *An iterative greedy algorithm*, bottom of page 419 and page 421)
- *Note*: we did not discuss in detail the intermediate step (subsection *A recursive greedy algorithm*)
- we worked out how the greedy algorithm works on the example of page 415, giving solution $\{a_1, a_4, a_8, a_{11}\}$
- there are many other "greedy" algorithms for activity selection, for example "always choose the shortest available acitivity" is not always optimal: consider activities $[0, 10), [9, 11), [10, 20)$
- knapsack problem: fractional and 0-1 versions (subsection *Greedy versus dynamic programming*): greedy is optimal in the first case, not always optimal in the second case
- please read Section 16.2, *Elements of the greedy strategy* (containing the knapsack problem)